

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IX
JAVA DATABASE**



Oleh:

Ramadhan Wijaya

2211102208

IF-10-M

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

1. API untuk aplikasi GUI java

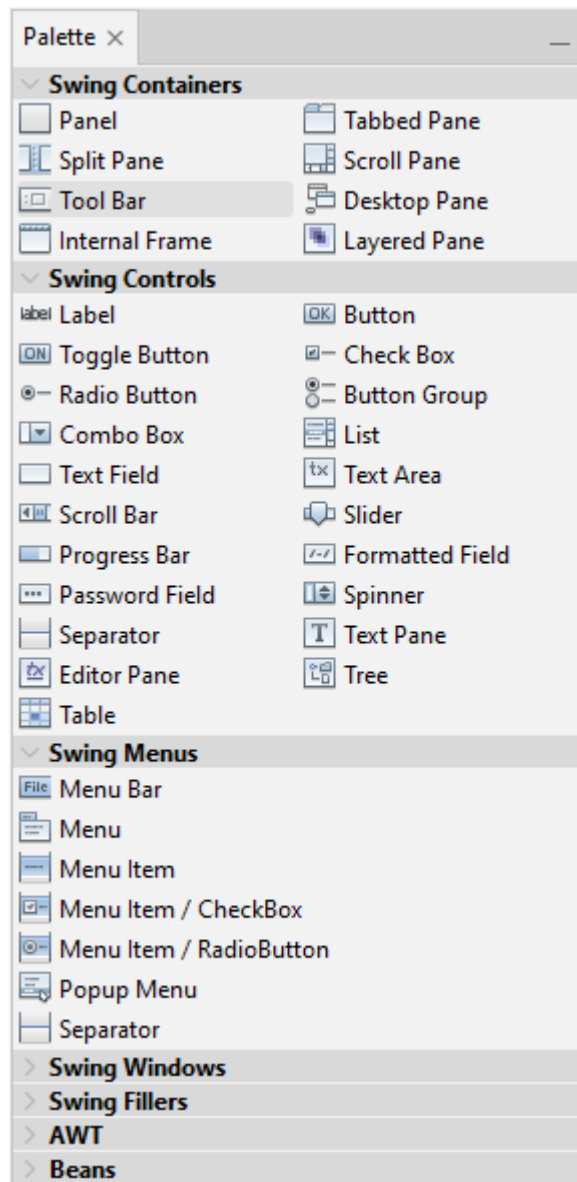
API atau application programming interface adalah seperangkat alat dan protocol yang digunakan untuk membuat perangkat lunak pada kondisi ini di java application. API menyediakan fungsionalitas untuk membuat antar muka grafis atau GUI(graphical User Interface). pada java application API yang klasik digunakan adalah java AWT(abstract window toolkit) API, Swing API, dan javaFX API. Berikut adalah penjelasannya :

- Java AWT
java AWT adalah toolkit GUI asli untuk java, namun karena perkembangan zaman java AWT digantikan oleh Swing dan javaFX karena keterbatasannya dalam fleksibilitas dan tampilannya yang lebih menarik. komponen AWT termasuk frame, Button, Label, TextField, dan TextArea.

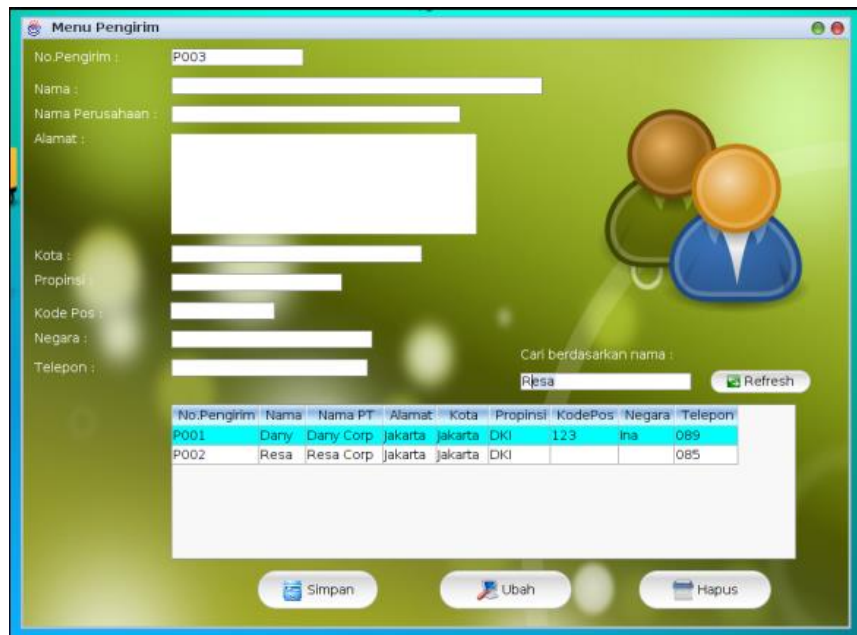
Berikut adalah contoh sederhana aplikasi yang menggunakan java AWT menggunakan grid LayOut.



- Java Swing
Swing adalah bagian dari java foundation classes (JFC) yang digunakan untuk membangun aplikasi GUI di java, sama halnya dengan AWT. Swing menyediakan komponen GUI yang lebih kaya dari AWT dan lebih mudah disesuaikan. Berikut adalah komponen komponen yang disediakan oleh java Swing



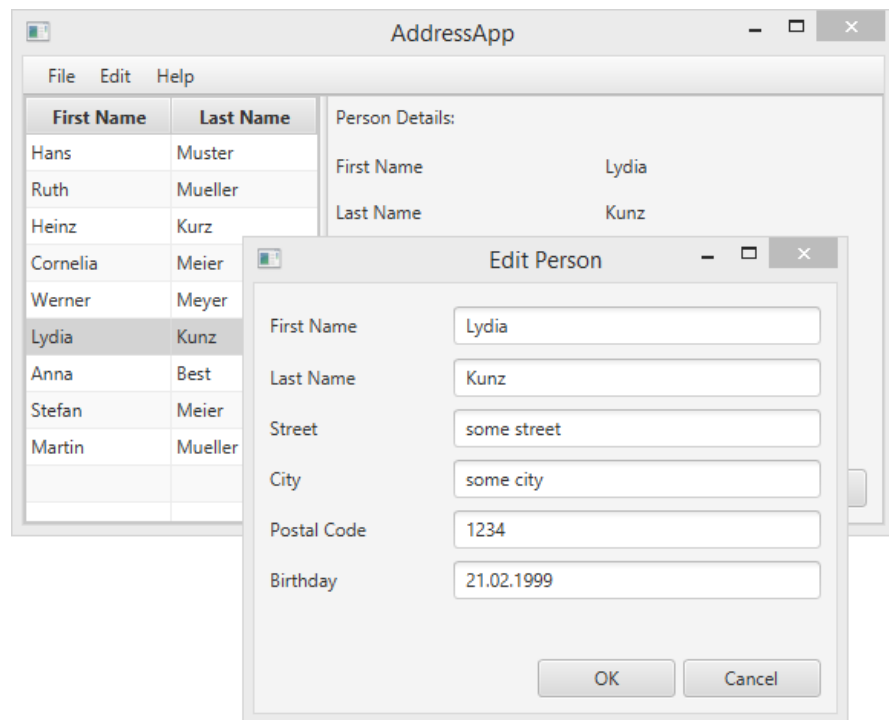
Berikut adalah contoh sederhana GUI dari aplikasi yang dibuat dari Java Swing



Teradapat table di bawah keterangan pengirim yang Dimana table tidak tersedia di java AWT.

- **JavaFX**
javaFX adalah framework yang lebih modern untuk membangun aplikasi di java GUI. javaFX dirancang untuk menggantikan Swing dan menawarkan lebih banyak fitur dalam pengembangan antarmuka pengguna yang lebih canggih. Cara kerja JavaFX melibatkan beberapa komponen utama dan konsep yang perlu dipahami untuk mengembangkan aplikasi GUI yang efektif.

Berikut adalah contoh aplikasi yang dibuat oleh JavaFX



2. JDBC dan JDBC API

JDBC (Java Database Connectivity) adalah sebuah API (Application Programming Interface) di Java yang memungkinkan aplikasi Java untuk berinteraksi dengan berbagai jenis sistem manajemen basis data (DBMS). JDBC menyediakan metode dan mekanisme standar untuk mengakses dan mengelola data yang tersimpan dalam basis data, sehingga pengembang dapat membuat aplikasi database-independen yang dapat bekerja dengan berbagai jenis DBMS.

JDBC API adalah sekumpulan kelas dan antarmuka dalam paket `java.sql` dan `javax.sql` yang menyediakan metode untuk berinteraksi dengan basis data. Komponen yang penting ada :

- **DriverManager:** Kelas ini digunakan untuk mengelola daftar driver basis data. Menggunakan metode `getConnection()` untuk membuat koneksi ke basis data.
- **Connection:** Antarmuka ini mewakili koneksi ke basis data. Menggunakan metode untuk membuat statement, mengelola transaksi, dan menutup koneksi.
- **Statement:** Antarmuka ini digunakan untuk menjalankan query SQL

Gambaran pengimportan JDBC

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
```

Dari connection sampai statement adalah komponen komponen dari JDBC

3. JDBC Components dan Architecture



Diatas merupakan gambar arsitektur relasi JDBC API dan JDBC Driver.

JDBC API berisi 2 set utama yaitu JDBC API untuk GUI dan JDBC API Driver untuk antarmuka database dari developer(tidak ditunjukkan kepada user).

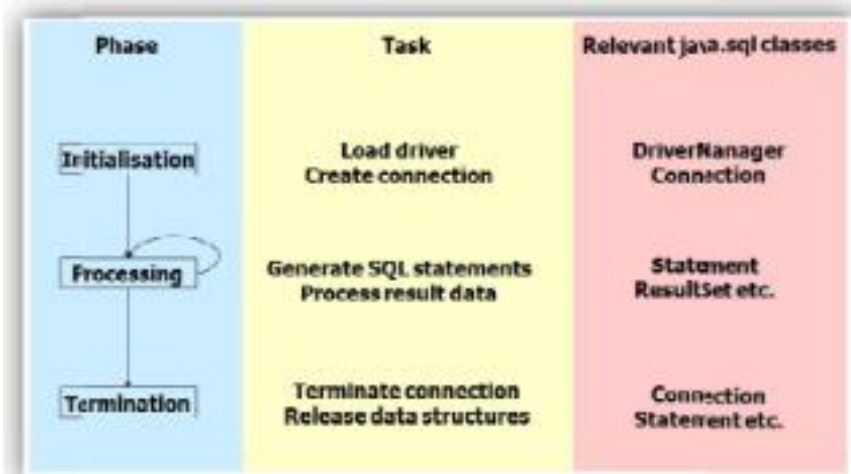
Classes	Function
DriverManager	Handle loading and unloading of drivers and establish a connection to a database
DriverPropertyInfo	All methods defined in this class are used to setup or retrieve properties of a driver. The properties can then be used by the Connection object to connect to the database
Type	The Type class is only used to define the constants used for identifying of the SQL types
Date	This class contains methods to perform conversion of SQL date formats and java Date objects
Time	This class is similar to the Date class, and it contains methods to convert between SQL time and java Time object
TimeStamp	This class provides additional precision to the java Date object by adding a nanosecond field

Diatas merupakan class class yang terdapat pada JDBC API.

Interfaces	Function
Driver	The primary use of the Driver interface is to create the Connection objects. It can also be used for the collection of JDBC driver meta data and JDBC driver status checking
Connection	This interface is used for the maintenance and status monitoring of a database session. It also provides data access control through the use of transaction locking
Statement	The Statement methods are used to execute SQL statements and retrieve data from the ResultSet object
PreparedStatement	This interface is used to execute precompile SQL statements. Precompile statements allow for faster and more efficient statement execution, and more important, it allows running dynamic query with querying parameters ' variation. This interface can be considered as a subclass of the Statement
CallableStatement	This interface is mainly used to execute SQL stored procedures. Both IN and OUT parameters are supported. This interface can be considered as a subclass of the Statement
ResultSet	The ResultSet object contains the queried result in rows and columns format. This interface also provides methods to retrieve data returned by an SQL statement execution. It also contains methods for SQL data type and JDBC data type conversion
ResultSetMetaData	This interface contains a collection of metadata information or physical descriptions associated with the last ResultSet object
DatabaseMetaData	This interface contains a collection of metadata regarding to the database used, including the database version, table names, columns, and supported functions

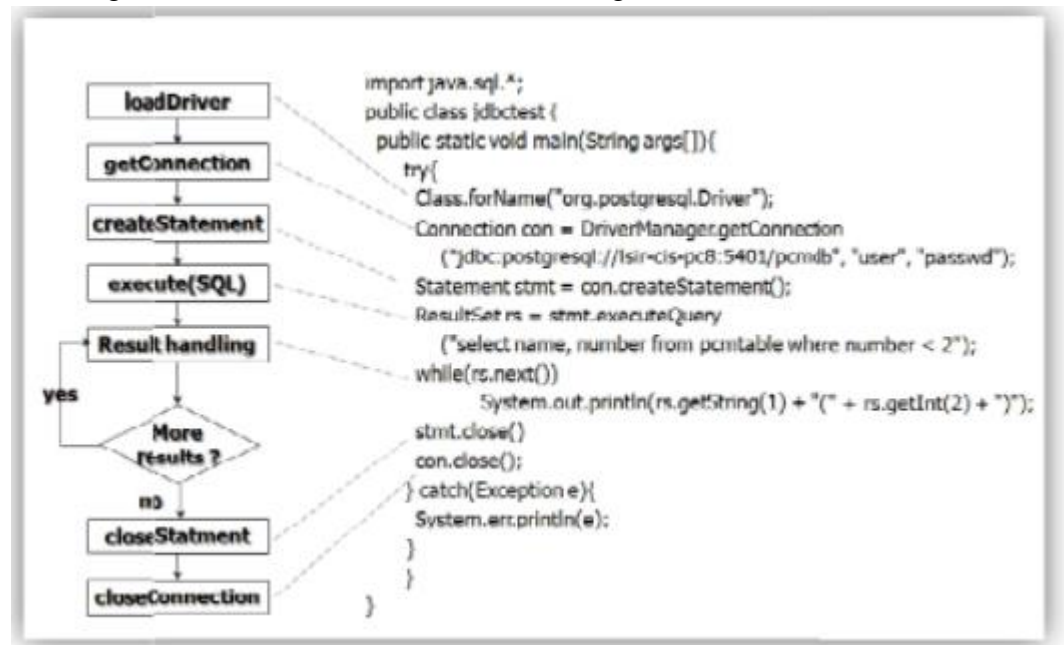
Sedangkan yang ini adalah interfaces yang terdapat pada JDBC API.

4. Cara Kerja JDBC

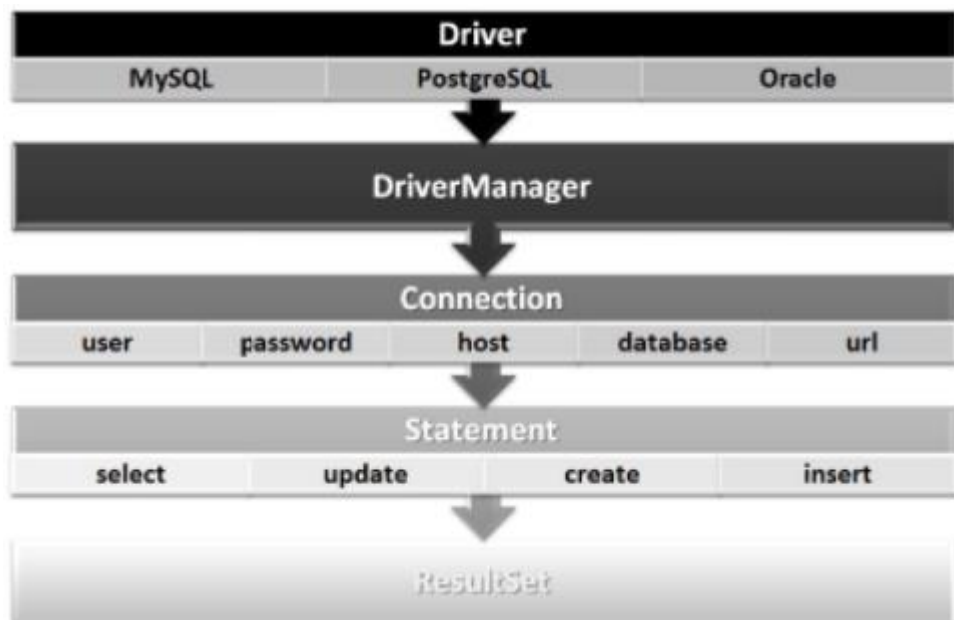


Diatas merupakan map yang menunjukkan cara kerja JDBC. Sederhananya pada saat memuat JDBC kita akan membuat koneksi, dan Driver manager pada tahap ini dilakukan di coding bagian Source. Selanjutnya ada tahap pemrosesan yang membuat dan mengeksekusi pernyataan SQL menggunakan "Statement" atau "PreparedStatement", dan terakhir menggunakan Resultset untuk memproses hasil querynya. Pada masa terminasi menutup semua resource yang digunakan (koneksi, statement, result set) untuk melepaskan memori dan sumber daya lainnya.

Berikut adalah gambaran membuat koneksi, membuat statement, result handling condition dan terakhir terminasi di bagian closeConnection.



5. Tahapan Akses Database

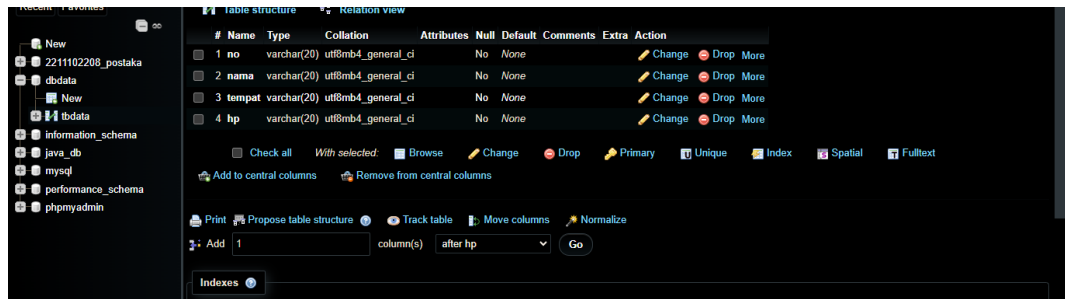


Driver memungkinkan java untuk berkomunikasi dengan basis data tertentu dari yang kita maksud. Dari banyaknya basisdata seperti PostgreSQL, oracle, MySQL. Selanjutnya ke driver manager, di driver manager ini kita akan mengoneksikan ke basis data. Setelah itu ke

connection untuk mewakili koneksi fisik ke basis data. Yang ke-4 ada statement untuk menjalankan pernyataan SQL terhadap basis data, pernyataan seperti select, insert, update, delete dll. Digunakan untuk mengelola hasil querinya. Terakhir masuk ke resultset, resultset adalah antarmuka yang mewakili hasil dari query SQL (yang ditunjukkan ke user). Selain itu resultset juga membaca proses hasil query yang dikembalikan oleh basis data ke user agar dapat melihat dengan jembatan antarmuka yang friendly dengan user.

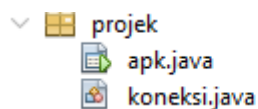
II. GUIDED

Pertama kita buat databasenya di myadminphp dengan XAMPP.

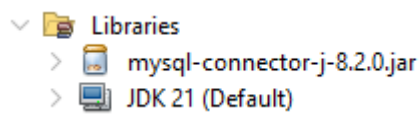


Diatas adalah datababase yang saya buat untuk guided. Jadi ada database “dbdata” yang memiliki table “tbdata”. Dalam tbdata terdapat ketentuan atribut atribut yang sesuai yang kita jalankan di praktikum.

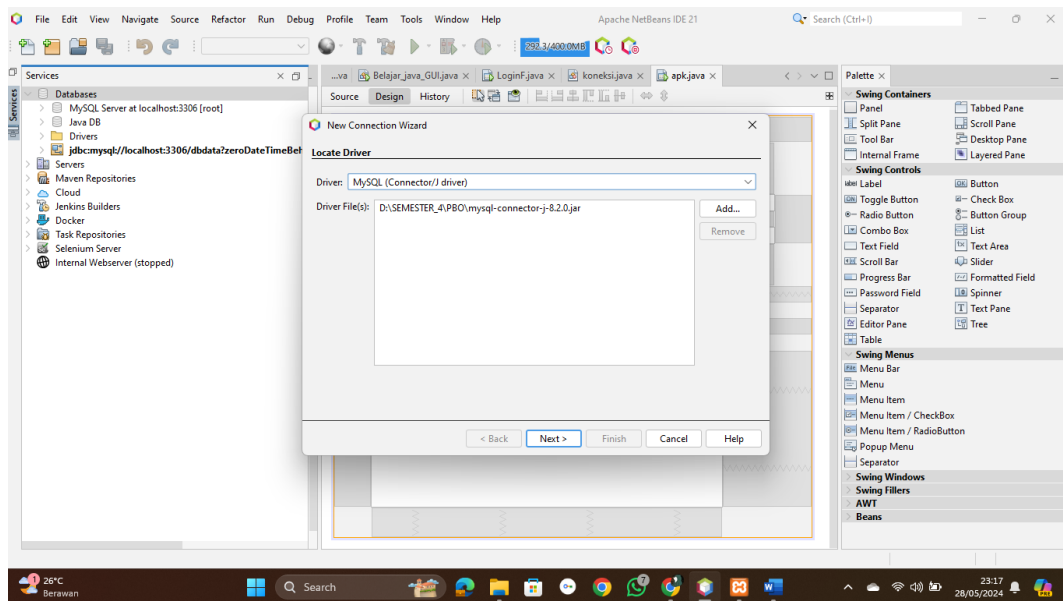
Setelah membuat database dan table yang sesuai ketentuan selanjutnya kita buka netbeansnya.




Dalam netbeasn kita membuat package projek yang didalamnya terdapat apk.java yang meruakan JFrame form dan koneksi.java yang merupakan java class



Selanjutnya pada folder libraries kita masukan mysql connector jarnya untuk mengedit GUI.



Selanjutnya kita masuk ke window->service. Lalu membuat menyambungkan phpMyAdmin dengan netbeansnya seperti diatas.

 **jdbc:mysql://localhost:3306/dbdata?zeroDateTimeBeha**

Ketika sudah akan muncul gambar seperti diatas di bagian service di dalam database.

```

/**
 *
 * @author rama
 * 2211102208
 */
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class koneksi{
    private static Connection mysqlkoneksi;
    public static Connection koneksiDB() throws SQLException{
        if(mysqlkoneksi == null){
            try{
                String DB = "jdbc:mysql://localhost:3306/dbdata";
                String user = "root";
                String pass = "";
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                mysqlkoneksi = (Connection) DriverManager.getConnection(DB,
user, pass);
            }
        }
    }
}

```

```

        }catch(Exception e){

        }
    }return mysqlkoneksi;
}
}

```

Selanjutnya kita masuk ke class koneksi.java untuk memasukan program diatas yang Dimana fungsinya adalah membuat koneksi dengan mengimport *import java.sql.Connection;*

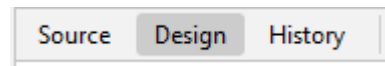
Setelah membuat koneksi ke phpMyAdmin sesuai gambar diatas kita selanjutnya membuat GUI untuk user di class APK.java

The screenshot shows a Java Swing window titled "JUDUL". Inside the window, there are four text input fields arranged in a 2x2 grid. The labels "no", "alamat", "nama", and "hp" are placed to the left of each field. Below the input fields, there are three buttons labeled "simpan", "edit", and "hapus". Further down, there is a search section consisting of a text input field and a button labeled "cari". At the bottom of the window, there is a table with four columns. The column headers are "Title 1", "Title 2", "Title 3", and "Title 4". The table body is currently empty.

Berikut adalah UI buatan saya sesuai pertemuan di kelas praktikum.

Selanjutnya kita akan membuat setiap table, fieldText, dan tombol menjadi responsive agar bisa digunakan oleh user.

Agar responsive kita perlu menambahkan program di bagian source di kiri atas Ketika di class APK.java.



Diatas klik bagian source di class APK.java

Setelah masuk ke bagian source kita selanjutnya memasukan code untuk setiap button, textfield, dan tablenya seperti dibawah ini programnya

```
package projek;

/**
 *
 * @author rama
 * 2211102208
 */
import java.sql.*;
import javax.swing.table.*;
import java.text.SimpleDateFormat;
import javax.swing.JOptionPane;
import com.mysql.cj.xdevapi.Table;

public class apk extends javax.swing.JFrame {

    /**
     * Creates new form apk
     */
    public apk() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the
    form.
     * WARNING: Do NOT modify this code. The content of this method is
    always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    public void tampil_data() {
        DefaultTableModel tabel = new DefaultTableModel();
```

```

        tabel.addColumn("NIM");
        tabel.addColumn("NAMA");
        tabel.addColumn("ALAMAT");
        tabel.addColumn("TELP/HP");
        try {
            java.sql.Connection conn = (java.sql.Connection)
projek.koneksi.koneksiDB();
            String sql = "select *from tbdata";
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            ResultSet rs = pst.executeQuery(sql);
            while (rs.next()) {
                tabel.addRow(new Object[]{
                    rs.getString(1),
                    rs.getString(2),
                    rs.getString(3),
                    rs.getString(4),});
            }
            jTable2.setModel(tabel);
        } catch (Exception e) {
        }
    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new apk().setVisible(true);
            }
        });
    }

    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        no = new javax.swing.JTextField();
        alamat = new javax.swing.JTextField();
        nama = new javax.swing.JTextField();
        hp = new javax.swing.JTextField();
        tb_simpan = new javax.swing.JButton();

```

```

tb_edit = new javax.swing.JButton();
tb_hapus = new javax.swing.JButton();
tb_cari = new javax.swing.JButton();
cari = new javax.swing.JTextField();
jScrollPane2 = new javax.swing.JScrollPane();
jTable2 = new javax.swing.JTable();

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportView(jTable1);

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel1.setText("JUDUL");

```

```

jLabel2.setText("no");

```

```

jLabel3.setText("nama");

```

```

jLabel4.setText("hp");

```

```

jLabel5.setText("alamat");

```

```

nama.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        namaActionPerformed(evt);
    }
});

```

```

tb_simpan.setText("simpan");
tb_simpan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tb_simpanActionPerformed(evt);
    }
});

```

```

});

tb_edit.setText("edit");

tb_hapus.setText("hapus");

tb_cari.setText("cari");
tb_cari.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tb_cariActionPerformed(evt);
    }
});

cari.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cariActionPerformed(evt);
    }
});

jTable2.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane2.setViewportViewView(jTable2);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(66, 66, 66)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(66, 66, 66)

```



```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel3)
            .addComponent(jLabel2))
        .addGap(12, 12, 12)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(no,
                javax.swing.GroupLayout.DEFAULT_SIZE, 169, Short.MAX_VALUE)
            .addComponent(nama))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel5)
            .addComponent(jLabel4))
        .addGap(33, 33, 33)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(alamat,
                javax.swing.GroupLayout.DEFAULT_SIZE, 136, Short.MAX_VALUE)
            .addComponent(hp))
        .addGap(47, 47, 47))
        .addGroup(layout.createSequentialGroup()
            .addComponent(tb_simpan)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tb_edit)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(tb_hapus)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addComponent(cari,
                javax.swing.GroupLayout.PREFERRED_SIZE, 223,
                javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
        .addComponent(tb_cari)
        .addGap(246, 246, 246))))
.addGroup(layout.createSequentialGroup()
        .addGap(47, 47, 47)
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(264, 264, 264))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(66, 66, 66)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
        .addComponent(jLabel2)
        .addComponent(jLabel5)
        .addComponent(no,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(alamat,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(20, 20, 20)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel3,
        javax.swing.GroupLayout.PREFERRED_SIZE, 22,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel4)
    .addComponent(nama,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(hp,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(53, 53, 53)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(tb_simpan)
    .addComponent(tb_edit)
    .addComponent(tb_hapus))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(cari,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(tb_cari))
    .addGap(19, 19, 19)
    .addComponent(jScrollPane2,
        javax.swing.GroupLayout.PREFERRED_SIZE, 202,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(38, Short.MAX_VALUE))
);

pack();
} // </editor-fold>

private void namaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cariActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        // TODO add your handling code here:
    }

    private void tb_simpanActionPerformed(java.awt.event.ActionEvent
    evt) {
        try {
            String sql = "insert into tbdata values("
                + no.getText() + ","
                + nama.getText() + ","
                + alamat.getText() + ","
                + hp.getText() + ")";
            java.sql.Connection conn = (java.sql.Connection)
projek.koneksi.koneksiDB();
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            pst.execute();
            JOptionPane.showMessageDialog(null, "Berhasil disimpan");
            tampil_data();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Gagal disimpan");
            System.out.println(e.getMessage());
        }
    }

    private void tb_cariActionPerformed(java.awt.event.ActionEvent evt) {

        try {
            java.sql.Connection conn = (java.sql.Connection)
projek.koneksi.koneksiDB();
            String sql = "Select * from tbdata where no=" + cari.getText() + "
or nama=" + cari.getText() + """;
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            ResultSet rs = pst.executeQuery(sql);
            if (rs.next()) {
                no.setText(rs.getString(1));
                nama.setText(rs.getString(2));
                alamat.setText(rs.getString(3));
                hp.setText(rs.getString(4));
                JOptionPane.showMessageDialog(null, "Data tbdata : "
                    + cari.getText() + " ditemukan, Apakah Anda Ingin
Mengedit atau Menghapus?");
            }
        } catch (Exception e) {
        }
        cari.requestFocus();
    }

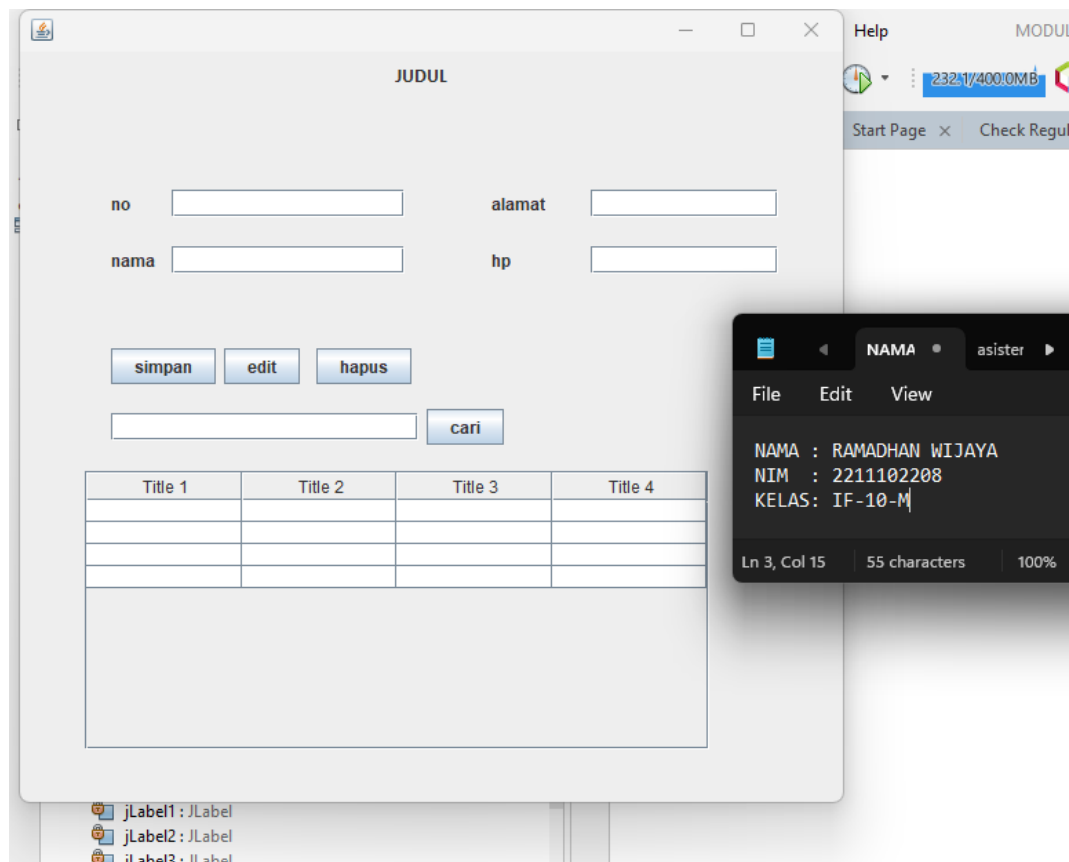
```

```
// Variables declaration - do not modify
private javax.swing.JTextField alamat;
private javax.swing.JTextField cari;
private javax.swing.JTextField hp;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JTable jTable2;
private javax.swing.JTextField nama;
private javax.swing.JTextField no;
private javax.swing.JButton tb_cari;
private javax.swing.JButton tb_edit;
private javax.swing.JButton tb_hapus;
private javax.swing.JButton tb_simpan;
// End of variables declaration
}
```

Program diatas menjelaskan komponen apa saja yang kita gunakan dalam mendesain GUI untuk class APK.java dan tempat memasukan fungsi untuk membuat setiap komponen komponen bisa responsive.

Setelah diisikan program diatas kita bisa langsung run di class APK.java di sourcenya

Berikut tampilan hasil runningnya



Dapat terlihat bahwa hasil runnya akan sesuai dengan apa yang kita desain. Program yang kita masukan tadi di APK.java akan membuat setiap komponen, sekarang saya coba masukan random data ke tabel tersebut apakah terhubung ke phpMyAdmin ?

JUDUL

no alamat

nama hp

Title 1	Title 2	Title 3	Title 4

wa hasil runnya ak
ang kita masukan ta

Berikut adalah data yang saya masukan

JUDUL

no alamat

nama hp

NIM	NAMA	ALAMAT	TELP/HP
1	mawl	karawarta	098312038
1	mawl	karawarta	098312038
1212	adadad	adadad	1212121
1222	hSdD	dDs	121212121
123121	12312312	awdsiknasId	as,dnlas
123121	12312312	awdsiknasId	as,dnlas
123121	12312312	awdsiknasId	as,dnlas
fdge	vn	ybbd	ivriyr
1	ucup	karawang	0123

Terlihat masuk di tampilan dari yang dikerjakan oleh program resultset

Berikut adalah database phpmyadminnya

Server: 127.0.0.1 » Database: dbdata » Table: tbdata

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)

[Profiling](#)
[\[Edit inline \]](#)
[\[Edit \]](#)
[\[Explain SQL \]](#)
[\[Create PHP code \]](#)
[\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

no	nama	tempat	hp
1	mawl	karawarta	098312038
1	mawl	karawarta	098312038
1212	adadad	adadad	1212121
1222	hSdD	dDs	121212121
123121	12312312	awdsilknasld	as,dnlas
123121	12312312	awdsilknasld	as,dnlas
123121	12312312	awdsilknasld	as,dnlas
fdge	vn	vbbd	jyrjyr
1	ucup	karawang	0123

☐ Show all | Number of rows: 25 | Filter rows:

Query results operations

Terlihat data “UCUP” sudah tersimpan di database phpMyAdmin.

Sekian guided kali ini... terimakasih!!